

**UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF ILLINOIS
EASTERN DIVISION**

| | | |
|---|---|-----------------------------------|
| EOLAS TECHNOLOGIES INC., |) | |
| |) | |
| and |) | |
| |) | |
| THE REGENTS OF THE UNIVERSITY OF CALIFORNIA, |) | |
| |) | |
| Plaintiffs, |) | |
| |) | |
| v. |) | No. 99 C 626 |
| |) | |
| MICROSOFT CORP., a Washington corp., |) | Judge Rebecca R. Pallmeyer |
| |) | |
| Defendant. |) | |

ORDER

Plaintiffs Eolas Technologies, Inc., and The Regents of the University of California won a substantial verdict on their claim that Defendant Microsoft Corporation infringed Plaintiffs' '906 patent. The Federal Circuit reversed the judgment on that verdict in part and remanded the case for a re-trial, ruling that the trial court had erred (a) in barring evidence relating to certain prior art and (b) in ruling that the patentees were not guilty of inequitable conduct. The Federal Circuit specifically upheld the trial court's claim construction. Microsoft nevertheless asked this court to reopen the matter of claim construction, but in an order dated April 12, 2007, this court declined to do so, citing the law-of-the-case doctrine and the mandate rule. Microsoft has moved for reconsideration of that ruling.

As explained in the court's April 12 Order, Microsoft argues that a re-evaluation of the '906 patent is required in light of a PTO reexamination of that patent, concluded after the Federal Circuit's decision. Microsoft asserts that the reexamination utilized a claim construction materially different from the one adopted by the courts. See April 12 Order, at 12. This court held that absent any showing that the PTO had in fact used a conflicting claim construction, this case presents no

“exceptional circumstance” that would allow a departure from the Federal Circuit’s mandate and a re-opening of claim construction on remand. *Id.* at 15; see *E-Pass Techs., Inc. v. 3Com Corp.*, 473 F.3d 1213, 1219 (Fed. Cir. 2007) (Federal Circuit’s claim construction “ordinarily remains the law of the case unless it is in conflict with a subsequent decision by [the Federal Circuit] sitting en banc”); see also *Nemmers v. United States*, 870 F.2d 426, 429 (7th Cir. 1989) (noting that only in “exceptional” circumstances can a district court, due to new and “substantially different” evidence, depart from the appellate court’s mandate).

The court did allow Microsoft to submit an offer of proof that in the reexamination proceeding, the PTO in fact allowed the ‘906 claims to survive based on a materially narrower claim construction. In this motion for reconsideration, Microsoft contends that it has made this showing, and that the court is now required to construe the ‘906 claims anew. (Microsoft’s Offer of Proof and Motion to Reconsider Regarding Revision of Claim Construction (“Def.’s Mot.”), at 1-2.) Because the evidence does not, in the court’s view, clearly establish that the PTO’s construction of the disputed terms in the ‘906 claims was in fact inconsistent with the claim construction articulated by the Federal Circuit, the court concludes that no departure from the law of the case is warranted.

A. The Technology at Issue

The ‘906 invention enables the user of a web browser, when visiting a web page, to interact with an “object,” such as a 3-D image, that is displayed within the browser window when the browser retrieves and renders the web page. The interactivity is enabled automatically, that is, without any further input from the user in the form of mouse clicks; the user does not have to navigate away from the browser window nor manually enable the interactivity. In general terms, the invention outlines a process in which the browser, when parsing the text-based HTML code of the web page, encounters a particular section of code, in an “embed text format”; that code indicates the location of an object external to the web page, e.g., the 3-D image; that object has “type information associated with it,” which indicates that an external “executable application,” or

program, is required to process the image; the browser then uses that “type information” to “identify and locate” the necessary application, which is automatically launched without any further input (i.e., mouse clicks) from the user; the image is then displayed in the browser window, where the user can interact with it. ‘906 Patent, col. 17, ll. 12-27; see *Eolas Techs., Inc. v. Microsoft Corp.*, No. 99 C 0626, 2000 WL 1898853, at *3 (N.D. Ill. Dec. 29, 2000) (initial claim construction order issued by Judge Zagel).

B. Brief Procedural History

Plaintiffs, the holders of the ‘906 patent,¹ filed a patent infringement suit against Microsoft in February 1999, claiming that certain aspects of Microsoft’s web browser Internet Explorer (“IE”) incorporated the ‘906 invention. Microsoft’s asserted defenses included invalidity and inequitable conduct, both related to the “Viola” browser invented by Pei-Yuan Wei. See *Eolas Techs. Inc. v. Microsoft Corp.*, 399 F.3d 1325, 1329 (Fed. Cir. 2005). at 1329. Specifically, Microsoft alleged that Viola was prior art that had been in public use more than a year before the ‘906 invention, thus anticipating it or rendering it obvious; and that Doyle, one of the ‘906 inventors, knew of Viola but failed to disclose any Viola reference to the PTO. *Id.* Judge Zagel, presiding over trial proceedings in this district, concluded as a matter of law that Wei had abandoned the version of Viola (DX34²) that allegedly was in public use. *Id.* He thus barred Microsoft from presenting evidence of DX34 to the jury. The judge did admit evidence related to a second version of Viola (DX37), but, before the case went to the jury, he granted Plaintiffs’ Rule 50 motion, ruling that DX37 did not, as a matter of law, anticipate or render obvious the ‘906 patent. See *id.* at 1329-30. On August 11, 2003, the

¹ The ‘906 patent, U.S. Patent No. 5,838,906, was issued on November 17, 1998 to inventors Michael Doyle, David Martin, and Cheong Ang. Plaintiff The Regents of the University of California owns the patent, and granted an exclusive license to Plaintiff Eolas Technologies.

² DX34, labeled as such by Microsoft as an exhibit in the first trial, is an archived file containing Viola source code; it is dated May 12, 1993. The version labeled DX37 is dated May 27, 1993. See *Eolas Techs.*, 399 F.3d at 1329.

jury found that Microsoft had infringed and had actively induced United States users of IE to infringe certain claims of the '906 patent, and had not shown that the patent was invalid. Special Verdict, Dkt. No. 475. The jury awarded a royalty of \$1.47 per unit of infringing product sold, i.e., Windows with IE, including foreign sales, for total damages of more than \$520 million. *Id.*; *Eolas Techs.*, 399 F.3d at 1332. After the jury returned its verdict, Judge Zagel conducted a bench trial on Microsoft's inequitable conduct defense. The court determined from the evidence at the bench trial that Doyle's limited knowledge of Viola did not trigger an obligation to bring that art to the PTO's attention, and that Viola was not prior art in any event. *Id.* at 1330; Ruling on the Defense of Inequitable Conduct, Dkt. No. 491, at 10-11.

Microsoft appealed, arguing, *inter alia*, that its Viola-related invalidity defenses should have been considered by the jury, that the district court erroneously construed the '906 claims, and that a new trial was required on the issue of inequitable conduct. (Brief of Appellant-Defendant Microsoft Corporation, Ex. K to Pl.'s Resp.) In an opinion issued on March 2, 2005, the Federal Circuit upheld Judge Zagel's claim construction (discussed below), as well as his conclusion that a royalty should include foreign sales.³ *Eolas Techs.*, 399 F.3d at 1336-41. The appellate court disagreed with Judge Zagel as to Microsoft's invalidity defenses, however, holding that the district court erred in finding as a matter of law that DX34 had been abandoned, that Wei's demonstration of Viola to two Sun Microsystems engineers did not constitute a public use, and that DX37 did not anticipate or render obvious the '906 patent. *Id.* at 1335. The Federal Circuit also vacated the district court's inequitable conduct ruling, which had relied in part on a finding that Viola did not constitute prior art. *Id.* at 1336. The case was remanded for further proceedings only as to Microsoft's anticipation and obviousness defenses, and its inequitable conduct defense. *Id.* at

³ Although the instant motion does not address the issue of foreign sales, the court recognizes that the Supreme Court's recent decision in *Microsoft Corp. v. AT & T Corp.*, 127 S. Ct. 1746 (2007), likely requires a different result on this issue.

1341. Upon remand, Microsoft demanded the case be reassigned to a different judge pursuant to a local rule. The Federal Circuit ultimately granted this request, *see Eolas Techs. v. Microsoft Corp.*, 457 F.3d 1279, 1281 (Fed. Cir. 2006), and the action was assigned to this court on October 6, 2006.

C. PTO Reexamination

As Judge Zagel noted in his ruling on claim construction, the PTO had rejected the '906 application three times before issuing the patent, each time for obviousness in light of the teachings of prior art.⁴ *See Eolas Techs.*, 2000 WL 1898853, at *9-12. That prior art did not, however, include Viola. On October 30, 2003, shortly after trial, the Director of the PTO ordered a reexamination of the '906 patent (the "Reexamination"), in which the PTO considered the DX37 Viola reference, that is, the second version of Viola, as prior art. In September 2005, after the Federal Circuit's decision, the PTO Examiner concluded that Viola did "not teach nor fairly suggest" the '906 invention. (Examiner's Statement of Reasons for Patentability and/or Confirmation (the "Reasons"), at 60, 61.)

The court briefly notes the parties' dispute with respect to the circumstances surrounding the Reexamination. According to Plaintiffs, Microsoft "instigated" the proceeding, following the adverse jury verdict, through a "campaign to get the Director of the PTO to initiate a reexamination" based on two prior art references ("Raggett I" and "Raggett II") that had been admitted as evidence in the trial. (Pl.'s Resp., at 3.) Microsoft denies initiating the proceeding. (Def.'s Reply, at 13.) Neither side offers evidence in support of these assertions, and the court finds the dispute irrelevant for purposes of the pending motion. Plaintiffs next assert that they, rather than Microsoft,

⁴ The prior art that rendered the '906 patent obvious, in the Examiner's opinion, included the Mosaic browser—which is unpatented—in combination with basic web technologies; a combination of Mosaic and the "Khoyi" patent, licensed to Wang Laboratories, Inc., which taught an operating system; and Microsoft's "Koppolu" patent, which taught OLE linking and embedding, in combination with other disclosed prior art including Mosaic. *See Eolas Techs.*, 2000 WL 1898853, at *9-11.

immediately submitted the DX34 and DX37 Viola references to the Examiner, and that Microsoft “made no effort” to do so. (Pl.’s Resp., at 3-4.) Microsoft calls this characterization misleading, asserting that it did not put Viola before the PTO because Judge Zagel had determined that Viola was not prior art; this finding, according to Microsoft, would have precluded the Examiner from considering Viola. (Def.’s Reply, at 13.) See 35 U.S.C. § 302 (allowing “[a]ny person” to file request for examination “on the basis of any prior art”); 35 U.S.C. § 301 (prior art consists of “patents or printed publications”). Again, this dispute is not relevant, although Plaintiffs’ factual assertion appears accurate: on December 30, 2003 an attorney for the patentees submitted the DX34 and DX37 Viola source code to the Examiner, noting that these references had been “asserted . . . as being prior art” in this litigation, and requested that the references “be expressly considered” during the Reexamination. (Information Disclosure Statement for Related Litigation, Ex. A to Pl.’s Resp.)

As Microsoft points out, the first Examiner assigned to the Reexamination initially declined to consider the Viola code submitted by the patentees because the patentees had not admitted that DX34 and DX37 in fact constituted prior art; rather, in the words of the Examiner, the patentees had “merely pointed to a ruling of a U.S. District Court that raises questions as to whether the Viola source code was publicly available.” (Information Disclosure Statement, Ex. B to Pl.’s Resp.) In an “Interview Summary” dated August 15, 2005, however—after the Federal Circuit’s decision on appeal—a different Examiner noted that the Office of Patent Legal Administration was considering whether Viola should be considered a “publication” and thus should be reviewed as prior art in the reexamination proceedings. (Interview Summary, Ex. C to Pl.’s Resp.) According to Microsoft, it had at that time no ability to submit information in the *ex parte* proceedings. (Def.’s Reply, at 14.) Microsoft asserts that Plaintiffs had this ability, but that Plaintiffs withheld (unidentified) “key documents” related to Viola, (Def.’s Reply, at 14), an assertion the court need not address here.

Ultimately, the PTO determined that the DX37 Viola code qualified as a publication that

constituted prior art,⁵ and the Examiner considered the DX37 code among the prior art references reviewed during the Reexamination. (Reasons, at 45.) As noted, the '906 claims survived the Reexamination; in the 73-page Reasons, the Examiner explained why none of the prior art references considered—including Viola, Berners-Lee, Raggett I & II, and Toye—invalidated the '906 patent. Although this outcome was not favorable to Microsoft, Microsoft now, as discussed more fully below, attempts to turn defeat at the PTO into victory in this litigation: according to Microsoft, the Reasons demonstrate that the Examiner construed several key terms of the '906 claims significantly more narrowly than those terms have been construed throughout this litigation; and under that construction, Microsoft maintains, its products would not infringe the '906 patent.

According to Plaintiffs, Microsoft has attempted to initiate yet another reexamination of the '906 patent. (Pl.'s Resp., at 5.) Plaintiffs point to a request for an *ex parte* reexamination, submitted to the PTO on December 22, 2005 by a law firm, Klarquist Sparkman, LLP, which, Plaintiffs assert, has represented Microsoft in other matters.⁶ (*Id.*; Request for *Ex Parte* Reexamination Transmittal Form, Ex. H to Pl.'s Resp.) See 37 C.F.R. § 1.510 (allowing “[a]ny person” to “file a request for an *ex parte* reexamination by the [PTO] of any claim of the patent on the basis of prior art patents or printed publications”). Neither party has apprised the court of the status of that request.

D. Claim Construction

⁵ Although it is not immediately apparent how the PTO came to this conclusion, the Reasons contain a quoted passage from the Federal Circuit’s decision on appeal, in which that court stated that the trial record shows that Wei had posted DX37 on an Internet site and notified a Sun Microsystems engineer—Wei had previously demonstrated DX34 to two Sun employees—that DX37 was available for download. (Reasons, at 45.) See *Eolas Techs.*, 399 F.3d at 1333. This court assumes that the PTO considered this a “publication.”

⁶ Indeed, the firm’s website identifies Klarquist Sparkman (which has not filed an appearance in this case) as “Microsoft Preferred Legal Counsel” and states that the firm “proudly served as patent prosecution and litigation counsel for Microsoft Corporation since 1991.” See <http://www.klarquist.com/about.aspx?Show=4>.

As noted, the '906 patent describes an invention in which a web browser, parsing the HTML code in a web page, encounters embedded text that specifies the location of an object external to the web page, e.g., a 3-D image; the browser then automatically launches an “executable application” that processes the image, which is then displayed in the browser window where the user can interact with it. Claims 1 and 6 of the '906 patent read, in relevant part: “wherein said object has type information associated with it utilized by said browser to identify and locate an executable application . . . [which the browser will] automatically invoke . . . in order to display said object and enable interactive processing of said object” ‘906 Patent, col. 17, ll. 15-23, col. 18, ll. 19-27. Microsoft identifies three aspects of this language as significant, for purposes of claim construction, in light of the Reexamination: the meaning of “executable application”; the meaning of “object”; and whether the “executable application” must both display the object and enable interactivity with it. (Def.’s Mot., at 5-10.) In its motion for reconsideration, Microsoft contends that the Reasons indicate that the PTO construed these three aspects of the '906 claims “far more narrowly” than as construed during this litigation, and that Plaintiffs, by declining to object to the Reasons, acquiesced to that narrower construction in order to allow the claims to survive the Reexamination. (*Id.* at 6-7, 14-18.) Microsoft asserts that the narrower construction would not ensnare the features of IE that the jury in the first trial found had infringed the '906 patent. (*Id.* at 4-5.) Microsoft thus contends that it would suffer an injustice if the court were to adhere to the claim construction articulated by Judge Zagel and the Federal Circuit, for then the '906 claims would have been construed one way by the PTO to allow the patent to survive, and another way by the courts to find that Microsoft had infringed. (*Id.* at 17-18.) See *Spectrum Int’l, Inc. v. Sterilite Corp.*, 164 F.3d 1372, 1379 (Fed. Cir. 1998) (“[c]laims may not be construed one way in order to obtain their allowance and in a different way against accused infringers.”) (quoting *Southwall Techs., Inc. v. Cardinal IG Co.*, 54 F.3d 1570, 1576 (Fed. Cir. 1995)). This injustice, Microsoft maintains, warrants an exception from the law-of-the-case doctrine.

Microsoft's argument necessarily depends on a comparison between the court's construction of the three aspects of the '906 claims that Microsoft identifies, and the Reasons produced by the Reexamination. Before undertaking this task, however, the court makes several preliminary observations. First, to the extent Microsoft suggests that the court should examine the Reasons as if it is construing the claims anew, the court declines to do so. According to Microsoft, the court should "carefully compare the Statement of Reasons against the prior claim construction, determine whether they are different, and determine whether, under applicable law, the PTO's construction . . . would be respected by the Federal Circuit." *Id.* at 24. In the court's view, this approach disregards the fact that the claim construction articulated by the Federal Circuit stands as the law of the case. Both the Federal Circuit and the Seventh Circuit have emphasized that a departure from the law of the case requires new and materially or substantially different evidence, and that a district court is free to disregard an appellate mandate only in exceptional circumstances. *See Intergraph Corp. v. Intel Corp.*, 253 F.3d 695, 698 (Fed. Cir. 2001); *Nemmers*, 870 F.2d at 429; *Smith Int'l, Inc. v. Hughes Tool Co.*, 759 F.2d 1572, 1576 (Fed. Cir. 1985). Microsoft thus bears the burden of showing, first, that the Reasons clearly establish that the PTO in fact used a significantly different claim construction; and, further, that the difference is material in the sense that an irreconcilable conflict has arisen between that claim construction and the claim construction articulated by the Federal Circuit, such that the outcome of this case would clearly be affected. Thus, slight differences will not suffice; and if the Reasons give rise only to competing inferences, suggesting that the Examiner may or may not have used a narrower claim construction, or that any difference may or may not be material, Microsoft has not met its burden. *See Johns-Manville Corp. v. Guardian Indus. Corp.*, 116 F.R.D. 97, 101-02 (E.D. Mich. 1987) (noting that in light of the "spirit of finality which is implicit in all judgments," a court faced with a request to depart from an appellate mandate "should be cautious in exercising its discretion to reopen proceedings based on newly discovered evidence. A mere suspicion of error will not suffice, and the court must be 'convinced

to a certainty' that error warranting review exists.") (quoting *Perkin-Elmer Corp. v. Computervision*, 732 F.2d 888, 901 (Fed. Cir. 1984)). Similarly, unless Microsoft can make this threshold showing, it would be premature for the court to speculate as to whether the PTO's allegedly different construction of claim terms "would be respected by the Federal Circuit"; to do so would be to effectively engage in a *de novo* review of claim construction, which would render meaningless the law-of-the-case doctrine and the mandate rule.⁷

Second, Microsoft faces an uphill climb in demonstrating, based merely on the Examiner's statements in the Reasons, that the PTO in fact utilized a significantly narrower claim construction in allowing the '906 claims to survive over Viola and the other prior art. As the court explained in the April 12 Order, nowhere did the Examiner explicitly construe any claim term, nor even suggest that he was applying a construction that differed from the construction used throughout this litigation; rather, Microsoft "urges that an inference of a narrower construction can be drawn from certain of the Examiner's statements, and [Plaintiffs'] failure to object thereto."⁸ See April 12 Order, at 15. Because Microsoft thus signaled its intent to rely on inferential argument, rather than direct evidence, to establish that the PTO had used a narrower construction, the court held that Microsoft had failed to make the necessary threshold showing that the Reasons constituted the type of

⁷ The court further notes, as it did in the April 12 Order, that even if the Reexamination in fact utilized a narrower construction of certain claim terms, that would not necessarily dictate the claim construction in this litigation. See April 12 Order, at 15 n.4. As the Federal Circuit explained in *Phillips v. AWH Corp.*, 415 F.3d 1303, 1317 (Fed. Cir. 2005), a patent's prosecution history, which would include a reexamination, is "less useful for claim construction purposes" than the patent's claim language or the written specification. Thus, if Microsoft were to meet its burden of showing that an exception to the law of the case is warranted, the court would consider the Reexamination, along with the rest of the prosecution history, after examining the claim language and the specification; the court could not simply adopt the PTO's construction.

⁸ PTO regulations allow a patentee to "file a statement commenting on the reasons for allowance within such time as may be specified by the examiner." 37 C.F.R. § 1.104(e). As noted *infra*, Microsoft contends that because the patentees did not file such a statement with the PTO, commenting on or objecting to the Reasons, a presumption arose that they acquiesced to all aspects of the Examiner's reasoning. (Def.'s Mot., at 15-16.)

materially different evidence that merited a departure from the Federal Circuit's mandate. *Id.* at 13. As discussed more fully below, the court concludes that Microsoft's offer of proof and motion for reconsideration indeed rely on the very kind of inferential argument that the court believes is insufficient.

Indeed, this lack of direct evidence alone is sufficient to distinguish the cases Microsoft cites in arguing that the "new evidence" exception to the law of the case should apply. In *Bandes v. Harlow & Jones, Inc.*, 826 F. Supp. 700, 703-04 (S.D.N.Y. 1993), for example, the Second Circuit affirmed the district court's ruling that the Sandinista Nicaraguan government's confiscation of the plaintiff's property was illegal because the plaintiff was not afforded basic due process rights. Between that decision and proceedings on remand for resolution of other issues, however, elections resulted in the Sandinistas' defeat, and the new Nicaraguan government issued decrees implementing procedures for the return of property confiscated by the Sandinistas. *Id.* at 704. The district court on remand held that these decrees constituted new evidence that warranted a departure from the law of the case. *Id.* at 706-07. The new evidence in *Bandes* thus directly conflicted with the precise reason for the earlier ruling, and did so without resort to inference or speculation. This evidence of election of new leaders and a reversal of the relevant practice has nothing in common with the asserted new evidence in this case. *See also Bridge v. U.S. Parole Comm'n*, 981 F.2d 97, 104 (3d Cir. 1992) (exception to law of the case applied where, after district court's sentencing of defendant who had pled guilty to firearms charge, Parole Commission received ATF report indicating that defendant had admitted to conducting deadly firebombing; district court had not considered this "highly probative evidence" when initially determining the severity level of the offense for purposes of parole eligibility).

Furthermore, the fact that the Examiner did not articulate a different claim construction, nor voice disagreement with the Federal Circuit's claim construction, takes on added significance because the Examiner was unquestionably aware of the claim construction in place throughout this

litigation. As Plaintiff points out, the patentees submitted to the Examiner several of Judge Zagel's rulings, including his claim construction order, as well as the Federal Circuit's opinion. (Letters from Krueger to Assistant Commissioner for Patents, Exs. E, F, & G to Pl.'s Resp.) It is readily apparent that the Examiner did review the Federal Circuit decision: as noted, a passage from that opinion, which appears to be the basis for the PTO's determination that Viola was prior art, is quoted in the Reasons. (Reasons, at 45.) Finally, Plaintiffs assert that during an interview with the Examiner on August 18, 2005, the patentees submitted, as one of a set of slides, a slide entitled "Scope of the Claims" that stated: "The scope of patent terms used herein are as utilized during the District Court trial and as affirmed by the Court of Appeals for the Federal Circuit in its 2005 decision." (Ex. D to Pl.'s Resp.) This assertion is well-supported: Plaintiffs attach a copy of this slide to an interview summary, written by the patentees' attorney and submitted to the PTO, which notes "a set of slides which are attached hereto[.]" (*id.*), and a declaration from one of Plaintiffs' attorneys, stating that the "Scope of the Claims" slide was submitted during that interview. (Martinez Decl. ¶ 7.)

With this background in mind, the court addresses each of the three aspects of the '906 claims that Microsoft contends were construed more narrowly in the Reexamination.

1. "Executable Application"

The Federal Circuit agreed with Judge Zagel's definition of "executable application" as "any computer program code, that is not the operating system or a utility, that is launched to enable an end user to directly interact with data." *Eolas Techs.*, 399 F.3d at 1336. In *Markman* proceedings, Microsoft had contended that "executable application" referred only to "standalone programs." *Eolas Techs.*, 2000 WL 1898853, at *5. Microsoft defined a "standalone program" as one that can be executed, (i.e., launched, run, or started) independently, as opposed to one that functions only as a component of another program, such as a dynamically linked library (DLL) component.⁹ *Id.*

⁹ A popular Internet source of Windows information describes a dynamic link library
(continued...)

Both parties used the example of a spell checking DLL to illustrate a “component”: a user can run the spell checker while working in a word processing or spreadsheet program, but the spell checker does not (and cannot) run on its own. *Id.* In the ‘906 context, Microsoft argued that a component program, hosted by a web browser to perform some function, could not be the “executable application” referenced in the claims unless it could function independently of the browser. *Id.* Judge Zagel concluded that neither the claims, the specification, nor the prosecution history of the ‘906 patent supported Microsoft’s proposed limitation of the term to standalone programs. *Id.* at *6-14. Microsoft advanced the same argument in its appeal to the Federal Circuit, without success. *Eolas Techs.*, 399 F.3d at 1336.

In the current motion, Microsoft acknowledges that the scope of “executable application” as affirmed by the Federal Circuit encompasses components that run only as part of a web browser. (Def.’s Mot., at 3.) Microsoft explains the significance of this construction with respect to three software components that IE utilizes or supports: Active X controls, Java applets, and Netscape-style plug-ins. As a general proposition, these technologies, like the ‘906 invention, provide for the automatic display of and/or interaction with a data object in a web page. During trial, Plaintiffs had argued that IE’s support for these three technologies, that is, IE’s ability to utilize them to enable the kind of automatic display and interactivity claimed by the ‘906 patent, infringed the ‘906 patent. According to Microsoft, a definition of “executable application” that is limited to standalone programs would not encompass these technologies, and IE therefore would, contrary to the jury’s finding, not infringe the ‘906 patent. (*Id.* at 4.) Microsoft maintains that in the Reexamination, the Examiner, in distinguishing the ‘906 claims from certain aspects of the Viola browser, adopted a construction of “executable application” consistent with Microsoft’s contention that the term is limited to

⁹(...continued)
(DLL) as “a collection of small programs, any of which can be called when needed by a larger program that is running in the computer.” See SearchWinIT.com, http://searchwinit.techtarget.com/sDefinition/0,290660,sid1_gci213902,00.html.

“standalone programs.” (*Id.* at 7-8.)

Microsoft submits with its offer of proof the declaration of John P.J. Kelly, Ph.D., who describes the operation of the three IE-supported technologies noted above and further opines that those components would not be considered “executable applications” the way that term was construed in the Reexamination. The court here presents Dr. Kelly’s explanation of these technologies, without making findings of fact nor accepting Dr. Kelly’s opinions, before addressing Microsoft’s arguments concerning Viola and the Reexamination.

a. Active X, Java Applets, and Netscape-Style Plug-ins

Active X controls. Dr. Kelly describes an Active X control as “a software component based on Microsoft’s OLE [Object Linking and Embedding] architecture” that “enables invocation of ready-made components, including third-party components, that blend in and appear as integral parts” of IE and thereby extend IE’s “functionality.” (Kelly Decl. ¶ 8.) He explains that when IE parses a web page and encounters an OBJECT, APPLET, or EMBED tag,¹⁰ IE calls on the Windows operating system to locate and launch an Active X control. (*Id.* ¶ 10.) According to Dr. Kelly, the Active X control is loaded into the computer’s memory (RAM) in the same “address space” as IE; thus, the control “is merged” into IE and “is not a ‘separate application.’” (*Id.* ¶ 11 & fig. 1.)

Java applets. Java, a general purpose programming language developed by Sun Microsystems, can be used to create “applets” that are found on web pages. Dr. Kelly explains that applets exist in a form called “byte code,” which cannot be executed directly by a Windows computer; rather, an “interpreter” is required to translate the byte code into machine-readable binary code that can then be executed by the computer’s central processing unit (CPU). (*Id.* ¶¶ 12-13.) This allows computers with different operating systems, e.g., Macintosh or Linux, to run the program contained in the Java applet, provided that each computer has an interpreter to translate the

¹⁰ A “tag” is a word in the text-based HTML code of a web page that conveys a particular instruction to a web browser as the browser parses the code of the page.

applet's byte code into binary code. (*Id.* ¶ 13.) On Windows computers, that interpreter is Microsoft's Java Virtual Machine ("JVM").¹¹ (*Id.*) According to Dr. Kelly, the JVM is not a part of IE, but is implemented as an Active X control, (*id.* ¶ 14); therefore, when IE encounters an APPLET tag, Windows invokes the JVM just like any other Active X control. (*Id.* ¶¶ 15, 17.) As with Active X controls, Dr. Kelly opines that the JVM is "merged" into IE once invoked. (*Id.* ¶ 14.)

Netscape-style plug-ins. The Netscape web browser, released for public use in 1994, was widely used in the early days of the Internet. To play sound or video files found on web pages, users would download and install "helper applications" that opened in a second window when playing the file. A "plug-in" application, in contrast, is recognized automatically by the browser when parsing an HTML web page, and opens the audio or video file in the browser window. See SearchSMB.com, http://searchsmb.techtarget.com/sDefinition/0,290660,sid44_gci212800,00.html.

Dr. Kelly describes Netscape-style plug-ins as components that are launched as part of a web browser's "ongoing process" (i.e., the portion of the browser that is loaded into and remains in RAM while the browser program is running). (*Id.* ¶ 18.) When a plug-in is launched as a result of IE encountering the EMBED tag while parsing a web page, the plug-in, according to Dr. Kelly, again "merges" into IE's process in RAM. (*Id.*)

b. Viola as Construed by the Examiner

The Viola reference considered in the Reexamination, like the '906 invention, Active X controls, Java applets, and Netscape-style plug-ins, is, as a general proposition, designed to enable the automatic display of an interactive data object in a web browser window. The Examiner described the Viola browser, and distinguished its operation from the '906 invention, by discussing a representative Viola file included in the DX37 materials submitted by Plaintiffs. (Reasons, at 46-

¹¹ Pursuant to a settlement of litigation between Microsoft and Sun Microsystems, the Microsoft JVM is no longer available for download, and Microsoft will cease supporting the product after December 31, 2007. See MSJVM Transition FAQ, <http://www.microsoft.com/mscorp/java/faq.asp>.

47.) Its purpose is to display an interactive clock in the window of a Viola web browser. (*Id.* at 47-50.) According to the Examiner, the browser, when parsing the code in a web page, encounters two tags labeled “VOBJF”; the text between those tags specifies a directory path (i.e., a location on the computer’s hard drive) and the name of a file—a “Viola script”—residing in that directory. (*Id.* at 47.) The browser then retrieves the Viola script—here, “clock.v”—from the directory location. (*Id.*)

The Examiner identified several significant attributes of Viola scripts such as “clock.v.” First, because the file is written in script form, rather than binary code, it must be interpreted by the Viola browser and then translated or converted into binary code before it can be executed by the CPU. (*Id.* at 47, 55.) Because a Viola script does not exist in a form the CPU can recognize, the script cannot be an “executable application” from the perspective of the CPU. (*Id.* at 55.) The Examiner noted that even if a Viola script is “pre-compiled” into “intermediate byte-code form,” the script would still not be an “executable application” because the browser would still have to complete the final step of translating the byte-code into binary code. (*Id.* at 55-56.)

Second, as addressed more fully below, the Examiner found that a Viola script could not be the “object” contemplated the ‘906 invention. (*Id.* at 53-54.) A Viola script such as “clock.v,” according to the Examiner, is an actual program; in contrast, the ‘906 claims contemplated “data objects,” which the specification defined as “information capable of being retrieved and presented to a user of a computer system.” (*Id.* at 53 (quoting ‘906 Patent, col. 3, ll. 36-39).) The Examiner noted that the ‘906 patent also encompassed “self-extracting data objects,” but found that any “executable component” of that self-extracting object was limited to the actual self-extraction, i.e., the “unpacking” of compressed data to its original size.¹² (*Id.*)

¹² Although the Examiner did not elaborate as to self-extracting data objects, the court assumes the term refers to compressed data files such as “.zip” files.

Finally, in one paragraph at the conclusion of his sixteen-page discussion of Viola,¹³ the Examiner stated that even if Viola's script-interpreting process were construed as "executing an application," Viola still did not teach the '906 patent claims because the interpreting process "merges" the so-called executable application, i.e., the Viola script, into the browser itself. (*Id.* at 60.) Specifically, the Examiner stated:

Assuming *arguendo* that . . . 'interpreting a script' may be considered as equivalent to 'executing an application,' then the Viola script arguably becomes an integral component of the Viola browser that parses, interprets (i.e., translates), and executes each line of the script (or corresponding byte-code). In such case, the browser and the 'executable application' *merge* into one program, and therefore cannot meet the requirement for a *discrete* 'browser application' and a *discrete* 'executable application' as claimed by the ['906 patent].

(*Id.* at 60 (emphasis added).)

In Microsoft's view, this passage conclusively establishes that the Examiner construed "executable application" in two significantly narrow ways. First, Microsoft seizes on the Examiner's use of the word "discrete": according to Microsoft, the PTO allowed the '906 patent to survive "*expressly* on the understanding" than the invention contemplated "a discrete 'browser application' and a discrete 'executable application.'" (Def.'s Mot., at 7 (emphasis in original).) Microsoft urges that by "discrete," the Examiner meant "standalone"; indeed, Microsoft asserts that "[t]here is no difference between a 'standalone application' and a 'discrete application.'" (*Id.* at 8.) Second, Microsoft contends that the Examiner "understood Viola to employ a process through which the 'executable application' merges into the browser when running," and that because Active X controls, Java applets, and plug-ins similarly "merge" into the browser, the PTO "plainly understood" the '906 claims as not encompassing those technologies. (*Id.*)

The court finds neither line of argument persuasive. First, it is not at all evident that "discrete" is synonymous with "standalone" in the context of computer software applications;

¹³ The Examiner distinguished Viola on several other grounds not relevant here.

Microsoft cites nothing in support of this assertion, and the court can find no support for it in the record. As Plaintiffs point out, the Examiner nowhere used the word “standalone,” despite being privy to the rulings in this case and therefore to Microsoft’s use of that term in its arguments before Judge Zagel and the Federal Circuit. Moreover, the record does contain a reference to “discrete” that supports precisely the opposite of what Microsoft now urges: in rejecting Microsoft’s argument that “executable application” was limited to “standalone applications,” Judge Zagel in his *Markman* order used the word “discrete” when describing *non*-standalone software components such as spell-checkers.¹⁴ *Eolas Techs.*, 2000 WL 1898853, at *5.

The notion that “discrete” should be considered equivalent to “standalone” becomes even less plausible when taking into account Microsoft’s earlier definition of a “standalone program.” Microsoft, in arguing before Judge Zagel that “executable application” should be limited to “standalone programs,” defined the latter as a program that “can be executed (launched, run, or started) irrespective of whether any other programs have been launched or are running.” *Eolas Techs.*, 2000 WL 1898853, at *5 (citing Microsoft’s Initial Brief on Claim Construction Issues). Microsoft’s conception of a “standalone program” was thus directed to a software program that

¹⁴ Judge Zagel explained as follows:

Computer code is often bundled into *discrete components*. These components can perform specific functions and be used as building blocks for larger programs. Some components exist separately from larger applications, and are summoned to assist a larger program (sometimes called a host program) when needed. Dynamically linked libraries (DLLs) are types of components that can be shared by different applications to perform common functions. The example used by the parties is a spell checker. Both word processors and spreadsheet programs offer spell checking capability and can share one spell checking DLL that exists as a separate block of code from the larger programs. Components like DLLs must be invoked by some other application, they cannot be executed without some host. Thus, in the spell checker example, one cannot run a spell checker unless another application, the word processor or spreadsheet, is also running.

Eolas Techs., 2000 WL 1898853, at *5 (emphasis added). Judge Zagel thus understood “discrete” applications as including those, like spell-checkers, that cannot run independently and are therefore *not* “standalone applications.”

could function independently of the browser, unlike, for example, a DLL such as a spell checker that requires a “host” word-processing or spreadsheet program. The court can find nothing in the Reasons to suggest that the PTO viewed a “discrete” application as one that can be launched or that can function independently of a host program.¹⁵

In Plaintiffs’ view, the Examiner’s use of the word “discrete” suggests nothing more than the “unremarkable conclusion” that the ‘906 invention contemplated a browser and an executable application that are two different programs. (Pl.’s Resp., at 8.) Plaintiffs note that the language of the claims already makes this plain: the claims refer to a “browser” application that must “identify and locate an [external] executable application.” (‘906 Patent, col. 17, ll. 17-20, col. 18, ll. 18-22.) Thus, the very fact that the browser must find and launch the executable application precludes the two from being the same application; in this sense, according to Plaintiffs, the browser and the “executable application” have always been understood to be “discrete” from one another.

Microsoft contends, however, that the Examiner used the term “discrete” in describing the status of the browser and the executable application *after* the browser has invoked the executable application. According to Microsoft’s interpretation of the Reasons, the PTO found the key distinction between Viola and the ‘906 claims to be that the Viola script “merges” into the browser, while in the ‘906 invention the executable application and the browser “remain distinct *while running*.” (Def.’s Mot., at 8 & n.6 (emphasis in original); Def.’s Reply, at 7-8.) Microsoft concludes

¹⁵ Microsoft also points to a passage in the Examiner’s discussion of the Raggett I reference in which the Examiner notes that the “executable application” of the ‘906 claims must be a “*separate* application from the browser application.” (Reasons, at 8 (emphasis added).) Microsoft does not develop its argument beyond quoting this language, and for the same reasons that the court finds the word “discrete” to be non-synonymous with “standalone,” the term “separate” is insufficient as well.

Moreover, the appearance of “separate,” in context, is even less suggestive of “standalone.” As discussed more fully below, the Examiner distinguished Raggett I from the ‘906 invention on the basis that in the former, the browser itself provides the interactivity, whereas in the latter, the “interactive processing” is “enabled by” the executable application, which necessarily is “separate” from the browser. (*Id.*)

that the Examiner must, therefore, have construed “executable application” as excluding components that “merge” into the browser, such as, according to Dr. Kelly, Active X controls, Java applets, and plug-ins.

The court is not persuaded that this is the most reasonable interpretation of the Reasons. Despite Microsoft’s repeated emphasis of the words “when running” or “while running,” and its characterization of this purported requirement as “the heart of the issue here,” (Def.’s Reply, at 8), the Examiner never in fact used this or similar language; nor did he explicitly identify a specific point in time at which the browser and the “executable application” of the ‘906 claims are “discrete.” Moreover, Microsoft’s argument relies entirely on a brief, one-paragraph passage, appended to the Examiner’s extensive discussion of Viola, that was phrased merely as an alternative ground for distinguishing Viola. The Examiner’s principal reason for distinguishing Viola from the ‘906 invention was that the so-called “executable application” in Viola was not an executable application at all, but a script that required the browser to interpret and translate the script into binary code. The Examiner made the distinction between the “merger” process used by Viola, and the “discrete” applications in the ‘906 patent, *only* by “assuming *arguendo*” that “executing an application” might encompass the process by which a Viola browser interprets a Viola script. The court does not read this alternative ground as necessarily translating to other contexts, such as to the process by which IE invokes an Active X control or the JVM, or as expressing the definitive opinion of the PTO with respect to the scope of the claim term “executable application.”

Furthermore, and perhaps most importantly, the Examiner did not describe the “merger” of Viola scripts and the Viola browser in the same sense that Dr. Kelly described the “merger” of Active X controls, Java applets, and Netscape-style plug-ins into IE. The Examiner used “merge” in explaining that the Viola browser is forced to parse the Viola script and interpret or translate that script into binary code; the Viola script thus becomes an “integral” part of the browser. (Reasons, at 60.) Dr. Kelly, on the other hand, used “merge” in the very different sense that the processes,

created when Active X controls (including the JVM) and plug-ins are invoked, are then loaded into the same physical address space in RAM as IE's process. (Kelly Decl. ¶¶ 10-18.) Dr. Kelly's description of the "merger" of IE components into IE through shared RAM address space thus takes place in a wholly different context from the Examiner's discussion of the interaction between the Viola browser and Viola scripts.¹⁶

In sum, it is not evident, from the Reasons, that the PTO in fact utilized a materially narrower construction of the claim term "executable application" in upholding the '906 patent in the Reexamination. To the contrary, the Examiner's construction appears to be entirely consistent with the construction articulated by Judge Zagel and the Federal Circuit. Microsoft's interpretation of the Reasons can be supported only if the court were to accept somewhat strained inferences arising from the Examiner's use of the terms "discrete" and "merge," and disregard more plausible readings of that language, such as Plaintiffs' interpretation that the Examiner merely used "discrete" in the sense that the "executable application" contemplated by the '906 invention is not the same program as the browser. As explained above, the court is under no obligation to resolve inferences in Microsoft's favor: Microsoft, which asks the court to disregard the Federal Circuit's mandate and depart from the law of the case, bears a threshold burden of clearly establishing that the PTO's construction of "executable application" is irreconcilable with the way that term has been construed throughout this litigation. Because Microsoft has not done so, the construction of "executable application" that was affirmed by the Federal Circuit remains the law of the case.

¹⁶ The court further notes that even under Dr. Kelly's description, it is not apparent that IE components such as the JVM somehow cease to be "discrete" from IE once invoked. Although he asserts that the JVM process (the DLL file msjava.dll) is loaded into the same address space as IE's process, (Kelly Decl. Fig. 2), Dr. Kelly does not assert that the JVM process actually disappears once so loaded. More importantly, as noted, the Reasons give no indication that the Examiner considered the terms "discrete" and "merge" in the RAM context.

2. Object

Judge Zagel did not construe the term “object” in the *Markman* proceedings; nor did the Federal Circuit pass on a definition of the term. As Microsoft points out, Judge Zagel did briefly address the issue when denying Microsoft’s motion for summary judgment for noninfringement. See *Eolas Techs. v. Microsoft Corp.*, No. 99 C 0626, 2002 WL 31375531, at *15 (N.D. Ill. Oct. 18, 2002). Microsoft had argued, in response to Eolas’ claim that IE’s support for Java applets infringed the ‘906 claims, that applets are not “objects” because they contain program code which, in conjunction with the JVM, enable the display of an image on a web page. *Id.* Judge Zagel pondered this question: “Is the end product, namely the image projected on a web page itself, the only thing that can be considered an object, or can the object encompass that image’s coding as well?” *Id.* Based on the evidence before the court at the time, Judge Zagel concluded that he had no “clear answer” to the question. *Id.*

According to Microsoft, the fact that Judge Zagel declined to answer the question, while denying Microsoft summary judgment, establishes a claim construction in which the term “object” is not limited to an object consisting entirely of the data that makes up the image to be displayed; rather, the scope of the term encompasses an object that also includes the programming code required to process and display the image. (Def.’s Mot., at 4 & n.3.) In the court’s view, this characterization is perhaps too specific: Judge Zagel never explicitly articulated this construction, but rather declined to impose the limitation that Microsoft sought. Accordingly, so far as the court is aware, the term “object” has not been explicitly defined in this litigation, and thus appears to have no limitations at all.

Microsoft now contends that the PTO Reexamination imposed precisely the limitation that Microsoft had urged upon Judge Zagel: an “object” cannot contain program code. As noted, the Examiner found that a Viola script such as “clock.v” could not be the “data object” contemplated by the ‘906 invention because a Viola script is an actual program. (Reasons, at 53-54.) Specifically,

the Examiner concluded that “the scope of the claimed ‘906 external ‘object’ clearly does not read upon a high-level source code PROGRAM, such as Viola script, nor does it read upon an object in byte-code form.” (*Id.* at 53.) According to Microsoft, this statement indicates that the PTO construed “object” to exclude any file that contains “any program coding,” (Def.’s Mot., at 9); the object “ha[s] to be only data.” (Def.’s Reply, at 11.) Microsoft thus contends that Java applets, which according to Dr. Kelly contain programming code in byte-code form in addition to data, cannot be “objects,” and that IE’s support for Java applets through the JVM thus does not infringe the ‘906 claims. (Def’s Mot., at 9.)

Plaintiffs interpret the Examiner’s statements regarding objects as being far less sweeping, and limited to the Viola context. (Pl.’s Resp., at 15.) Plaintiffs note that the Examiner concluded that neither “Viola scripts” nor “*corresponding* byte-code forms” anticipated the ‘906 claims. (Reasons, at 52 (emphasis added).) Plaintiffs thus maintain that the PTO did not exclude *all* files containing byte code from the definition of “object,” but only those byte-code forms “corresponding” to Viola scripts. (Pl.’s Resp., at 15.) Given what Plaintiffs call “Viola’s own peculiar architecture,” Java applets are not necessarily thus excluded. (*Id.*) Plaintiffs further point out that the ‘906 claims require objects that contain data in any event, and assert that the Examiner rejected the notion of Viola scripts as objects because the scripts contained no data. (*Id.*)

In the court’s view, neither party’s interpretation of the Reasons is significantly more convincing than the other. The fact that the Examiner excluded files with “high-level source code programming” from a definition of “object” does not, as Microsoft contends, necessarily establish that a file with “*any* program coding” cannot be an “object”; read literally, only a specific type of programming triggers the exclusion. This reading is borne out by the Examiner’s identification of Viola script, and byte-code programming that “correspond[s]” to Viola script, as examples of the kind of programming that would preclude a file from being an “object.” Microsoft’s interpretation thus appears too broad. On the other hand, the court is less certain than Plaintiffs that the

Examiner rejected Viola scripts as objects on the basis that the scripts contained no data. While the Examiner did refer to the “clock.v” script as “a high-level source code program,” and noted that “in contrast,” the ‘906 “object” referred to “data objects,” the Examiner never explicitly stated that the “clock.v” file or any other Viola script consisted entirely of programming without data. In short, it is unclear whether the Examiner, by excluding Viola scripts and “corresponding” byte-code forms from the scope of an “object,” merely excluded those files consisting entirely of programming code; or whether he excluded all “mixed” files of code and data; or whether he excluded only those mixed files where the program code was Viola-like.

As Plaintiffs appear to concede, the Examiner, by excluding Viola scripts, may have imposed at least some limitation on the scope of the term “object”; indeed, given that the term heretofore had no limitations at all, even Plaintiffs’ interpretation of the Reasons as excluding only Viola scripts and/or similar byte code does represent a change from the court’s position, declining to determine that any such limitation was appropriate. The parties’ dispute, therefore, centers on whether the difference is material. Microsoft defines materiality with respect to Java applets: if the PTO’s construction of “object” would exclude such applets, then that construction is materially different because, according to Microsoft, IE’s support for applets would then not infringe the ‘906 patent. Microsoft urges that such a result would be inconsistent with the jury’s verdict on infringement, and that the allegedly narrower construction of “object” that emerged from the Reexamination thus constitutes materially different evidence because it would lead to a different outcome in this action.

An inquiry as to whether certain limitations on a claim term would lead to a different outcome presents some difficulty, as the court is potentially forced to consider questions of infringement that were conclusively decided by the jury and are now within the Federal Circuit’s mandate. Indeed, if Microsoft is correct that the PTO construed “object” in such a way as to exclude Java applets, then the court would be in the uncomfortable position of essentially predicting, *post-hoc*, what effect

that construction might have had on the jury's finding of infringement. The court need not undertake that inquiry, however, because the Reasons do not clearly indicate that Java applets are not "objects" in any event. As noted, the fact that the Examiner rejected Viola scripts and Viola-like byte-code forms does not lead inescapably to a conclusion that a file that contains *any* programming data cannot be an "object," as Microsoft contends. Plaintiffs have suggested other interpretations of the Reasons, such as that the Examiner intended only to exclude unique Viola scripts and byte-code forms similar to those Viola scripts, or that the reason those scripts were excluded is that such scripts consist entirely of programming code. Java applets would not be excluded as "objects" under these constructions. In the court's view, the interpretations advanced by Plaintiffs are at least as plausible as Microsoft's, and would have no effect on any determination of infringement.

Given that the Reasons are open to such different interpretations, and that Microsoft's interpretation is merely equally plausible at best, Microsoft has failed to clearly establish that the PTO in fact utilized a materially different claim construction with respect to the scope of the term "object." The court thus adheres to the earlier construction of the term, which, as noted, the court has declined to define with any precision.

3. Display and Enabling of Interaction by the Executable Application

In instructing the jury as to the scope of the claim term "display," Judge Zagel explained that "[o]nce the executable application is invoked, the object is displayed to allow interactivity. The display of the object may be done by the executable application, the browser, or the executable application and the browser together." Tr. at 3424. This instruction arose from the court's decision denying Microsoft's motion for summary judgment for noninfringement, in which Judge Zagel rejected Microsoft's position that in the '906 invention, "the *browser*, and *only* the browser,

perform[s] the display function.”¹⁷ See *Eolas Techs.*, 2002 WL 31375531, at *20, 23-25 (emphasis in original).

Now, Microsoft contends that it is the executable application, and only the executable application, that can display an object. (Def.’s Reply, at 10.) This about-face was triggered by a passage Microsoft quotes from the Reasons, discussed below, which, according to Microsoft, establishes that “the executable application must *both* display the object and enable interactive processing” with the object. (Def.’s Mot., at 10 (emphasis in original).) Citing Dr. Kelly’s declaration, Microsoft avers that the JVM neither displays data nor provides interaction; rather, it simply translates a Java applet’s byte code into binary code, while the applet itself controls the display of, and interaction with, data. (*Id.*; Kelly Decl. ¶ 51.) Thus, Microsoft contends, the JVM is not an “executable application,” and IE’s support for applets therefore does not infringe the ‘906 patent under the construction adopted by the PTO in the Reexamination. (Def.’s Mot., at 10.)

Microsoft relies on the following statements by the Examiner in the Reasons:

. . . claims 1 and 6 therefore require an operative coupling between the claimed “executable application” and the claimed “interactive processing” such that the claimed “interactive processing” must be enabled by an executable application” that meets five explicitly claimed requirements:

* * *

4. *The executable application must display the object* within the display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window.
5. *The executable application must enable interactive processing of the object* within the display area created at the first location within the portion of the first distributed hypermedia document being displayed in the first browser-controlled window.

¹⁷ Microsoft again argued in its Rule 50 motion for JMOL that Judge Zagel had erred by not limiting the display function of the ‘906 invention to the browser. Dkt. No. 504, at 10-12. After Judge Zagel denied the motion, see Dkt. No. 546, Microsoft declined to appeal this aspect of claim construction to the Federal Circuit. (Brief of Appellant-Defendant Microsoft Corporation, Ex. K to Pl.’s Resp., at 2.)

(Reasons, at 9-10 (underscore in original; italics added).)

Microsoft quotes only the portions of points 4 and 5 that the court has italicized; Microsoft omits everything after the word “object,” and, significantly, does not explain the context in which these statements were made. As Plaintiffs point out, the Examiner reached these conclusions in distinguishing the ‘906 claims from a four-way combination of Berners-Lee, Raggett I, Raggett II, and admitted prior art; this combination taught a method of embedding static, rather than interactive objects, in web documents. (*Id.* at 3.) Unlike the “ongoing real-time manipulation and control by the user of the object displayed” within the browser window, as claimed by the ‘906 invention, the rendering applications contemplated by the prior art merely generate a static image and then terminate. (*Id.* at 5.) The Examiner acknowledged that some static images are “interactive” in the sense that they contain active areas—“active maps”—that allow the user to click on these areas, triggering the browser to invoke a URL and thus change the image. (*Id.*) This process, however, forces the *browser* to provide the interactivity upon extra input from the user (i.e., the mouse click); in the Examiner’s opinion, this aspect of the prior art was sufficient to distinguish that art from the ‘906 invention, in which a separate “executable application” (i.e., not the browser) automatically provides the interactivity, without the user’s input, when the browser parses the web page and automatically locates and launches the executable application. (*Id.* at 5, 8.)

The court is not persuaded that the Examiner’s statements clearly establish that the PTO understood that the “object” of the ‘906 invention can be displayed *only* by an executable application. As Judge Zagel noted in his claim construction order, “[c]ontext is important.” *Eolas Techs.*, 2000 WL 1898853, at *13. By relying only on selected quotations of the Examiner’s statements, and by neglecting to engage in any discussion concerning the prior art that the Examiner was distinguishing from the ‘906 claims when he made those statements, Microsoft attempts to demonstrate that the point of those statements was to define an “executable application” as one that both displays an object and enables interactivity, so that a program that

does not perform both functions cannot be an “executable application.” This was not, however, the basis upon which the Examiner distinguished the four-way prior art combination from the ‘906 invention. As noted, that prior art did not anticipate the ‘906 claims because the prior art rendered a static, rather than an interactive, image. In the prior art reference, the image was displayed by “executable rendering applications,” (*id.* at 5); thus, the Examiner necessarily compared those executable applications to the “executable application” in the ‘906 claims. The point of distinction was not that the executable applications in the four-way combination did not display the images—they did—but that those executable applications, unlike the executable applications in the ‘906 invention, did not enable interactivity. Because the display function of the prior art invention was performed by the putative “executable application” in that prior art, the Examiner had no occasion to address the issue of whether the display function of the ‘906 invention *necessarily must* be performed by an executable application, or whether, as in Judge Zagel’s construction, the browser, or a combination of the browser and the executable application, could perform the function. In other words, the Examiner did not necessarily say that the object can be displayed *only* by an executable application; rather, his comments can reasonably be interpreted as requiring that *when* an executable application performs the display function, it must do so within the browser window—the language Microsoft omits—and it must then enable interactivity. None of this is inconsistent with Judge Zagel’s construction of the display function.

Nor can the court entirely disregard the fact that Microsoft’s current position—that only the executable application can perform the display function in the ‘906 invention—is the very opposite of the position Microsoft took before Judge Zagel, when it argued that only the browser, and *not* the executable application, could display the object. Plaintiff further notes that Microsoft could have appealed Judge Zagel’s rejection of that position to the Federal Circuit, but chose not to do so. Although Microsoft dismisses the inconsistency by declaring, “That is simply not the issue here,” (Def.’s Reply, at 10), the fact that Microsoft can so completely reverse its position, at this late stage

in the litigation, defeats the notion that the construction Microsoft now urges is inevitable.


E. Summary and Conclusion

The court concludes that Microsoft has not demonstrated that the '906 claims survived the PTO Reexamination only due to a materially narrower claim construction than was affirmed by the Federal Circuit. Because the court finds that Microsoft has failed to make a threshold showing that the PTO in fact utilized a materially narrower claim construction, the court need not address Microsoft's further argument that Plaintiffs acquiesced to a narrower construction by failing to object to the Examiner's statements in the Reasons,¹⁸ nor its argument, advanced in a supplemental filing, that the Supreme Court's recent decision in *KSR Int'l Co. v. Teleflex, Inc.*, 127 S. Ct. 1727 (2007) emphasized particular deference to the PTO.

Microsoft's Motion for Reconsideration (665) is denied.

ENTER:

Dated: July 17, 2007


REBECCA R. PALLMEYER
United States District Judge

¹⁸ Microsoft points to a 2000 amendment of 37 C.F.R. § 1.104(e) that deleted the following language: "Failure to file [a statement commenting on an Examiner's reasons for allowance] does not give rise to any implication that the applicant or patent owner agrees with or acquiesces in the reasoning of the examiner." Changes To Implement the Patent Business Goals, 65 Fed. Reg. 54604, 54633 (Sept. 8, 2000). The comments to the proposed new rule predicted that "the failure of an applicant to comment on damaging reasons for allowance would give rise to a presumption of acquiescence to those reasons, and the negative inferences that flow therefrom." (*Id.*) Microsoft contends that as a result of the 2000 amendment, the '906 patentees had an "obligation to object" if they disagreed with the Reasons; but because Plaintiffs "remained silent," they "thereby acquiesced in the PTO's . . . narrow claim construction." (Def.'s Mot., at 15-16.)

It is not clear whether the state of the law is as Microsoft contends, however; since the 2000 amendment, the Federal Circuit has continued to note that "the prosecution history may not be used to infer the intentional narrowing of a claim absent the applicant's clear disavowal of claim coverage, such as an amendment to overcome a rejection." *Amgen Inc. v. Hoechst Marion Roussel, Inc.*, 314 F.3d 1313, 1327 (Fed. Cir. 2003). In any event, because Microsoft has not shown that the PTO in fact utilized a "narrow claim construction," the court need not reach the issue.